CLAIMS

1. A system for scanning for malware a computer file containing source code of a computer program in a given computer language comprising:

means for separating the source code into groups of constituent parts corresponding to different structural parts of the program;

means for processing each part to count the number of occurrences in that part of characters of a character set to obtain a frequency distribution of characters in that part;

means for comparing the character frequency distribution of each part with an expected range of frequency distributions; and

means for flagging the file as suspect or not depending on the result of one or more comparisons by the comparing means.

2. A system according to claim 1 wherein the flagging means is operative to flag the file as suspect if the comparing means detects that the frequency distribution of one or more of said parts does not match an expected range.

3. A system according to claim 1 wherein the flagging means is operative to flag the file as suspect depending on an accumulated score prepared by adding individual scores obtained in comparing each part with an expected frequency distribution.

4. A system according to claim 1, 2 or 3 wherein, in operation of the comparing means, the range of distributions which it considers as representing an acceptable match for the part is varied depending on the number of characters either in part or the program as a whole, with fewer characters corresponding to a wide range.

5. A system according to any one of the preceding claims and including:

means for maintaining an exception list of files which by their contents are to be treated as exceptions;

means for identifying a file as being included in the exception list; and

wherein a file is not marked as suspect if it is identified as being on the exception list.

6. A system according to any one of the preceding claims wherein duplicates of parts are ignored.

7. A method for scanning for malware a computer file containing source code of a computer program in a given computer language comprising:

separating the source code into groups of constituent parts corresponding to different structural parts of the program;

processing each part to count the number of occurrences in that part of characters of a character set to obtain a frequency distribution of characters in that part;

comparing the character frequency distribution of each part with an expected range of frequency distributions; and

flagging the file as suspect or not depending on the result of one or more comparisons by the comparing means.

8. A method according to claim 7 wherein the flagging means is operative to flag the file as suspect if the comparing means detects that the frequency distribution of one or more of said parts does not match an expected range.

9. A method according to claim 7 wherein the flagging means is operative to flag the file as suspect depending on an accumulated score prepared by adding individual scores obtained in comparing each part with an expected frequency distribution.

10. A method according to claim 7, 8 or 9 wherein, in operation of the comparing means, the range of distributions which it considers as representing an acceptable match for the part is varied depending on the number of characters either in part or the program as a whole, with fewer characters corresponding to a wide range.

11. A method according to any one of claims 7 to 10 and including:

maintaining an exception list of files which by their contents are to be treated as exceptions;

identifying a file as being included in the exception list; and

wherein a file is not marked as suspect if it is identified as being on the exception list.